This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/826,987	04/05/2001	David Bruce Kumhyr	AUS920000406US1	4117

35525 7590 07/02/2004

IBM CORP (YA)
C/O YEE & ASSOCIATES PC
P.O. BOX 802333
DALLAS, TX 75380

EXAMINER	
BLACKWELL, JAMES H	
ART UNIT	PAPER NUMBER
2176	

DATE MAILED: 07/02/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/826,987	**Applicant(s)** KUMHYR ET AL.	
	Examiner James H Blackwell	**Art Unit** 2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *05 April 2001*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-51* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-51* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *05 April 2001* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date *07/16/01*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-6, 14, 18-23, 31, 35-40, and 48 are rejected under 35 U.S.C. 102(b) as being anticipated by Hinks et al. (hereinafter Hinks, U.S. Patent No. 5,678,039).

In regard to independent Claim 1 (and similarly independent Claims 18, and 35), Hinks teaches that the Translation Table contains all the information needed to build editors that can simulate the target UI (User Interface) without having access to the sources or actual binary of the product to be translated (Col. 7, lines 22-26). In addition, Hinks teaches that the Translation Table encapsulates all the information that is known or can be derived from the various resources and stores them in a format which may be utilized by various editors (Col. 3, lines 16-19; compare with Claim 1 (and similarly Claims 18, and 35), ***"... receiving contextual information describing a visual context for the computer program's generation of the text; receiving the text from a first user interface text storage format; combining the contextual information with the text to form a visual representation of a display the user interface; and displaying the visual representation"***).

In regard to dependent Claim 2 (and similarly dependent Claims 19, and 36), Hinks teaches that once the end-user translator has completed the task of translating the resources, the translated text is merged back to sources. The Export/Import module is again employed, this time for generating a translated resource file. The translated resource file is similar to the original resource file, except that any necessary translations (e.g., translating an English text string into a French text string) have been carried out (Col. 3, lines 25-31; compare with Claim 2 (and similarly Claims 19, and 36), ***"... receiving editing instructions text; and based on the editing instructions, generating new text"***).

In regard to dependent Claim 3 (and similarly dependent Claims 20, and 37), Hinks teaches that once the end-user translator has completed the task of translating the resources, the translated text is merged back to sources. The Export/Import module is again employed, this time for generating a translated resource file. The translated resource file is similar to the original resource file, except that any necessary translations (e.g., translating an English text string into a French text string) have been carried out. In addition to translating text strings, other graphical user interface modifications, such as resizing of resources, have also been carried out (Col. 3, lines 25-34; compare with Claim 3 (and similarly Claims 20, and 37), ***"... storing the new text in a second user interface text storage format"***).

In regard to dependent Claim 4 (and similarly dependent Claims 5, 21-22, and 38-39), Hinks teaches that a (Export/Import) Resource Parser (330) parses the resource file (325) (such as a Windows .rc file) into a Translation Table (340), which is typically stored as a database table. The Translation Table (340) encapsulates all the information that is known or can be derived from the various resources and stores them in a format which may be utilized by various editors (Col. 7, lines 64-67; Col. 8, lines 1-5; compare with Claim 4 (and similarly Claims 5, 21-22, and 38-39), ***"... the first user interface text storage format is a catalog file/resource bundle"***).

In regard to dependent Claim 6 (and similarly dependent Claims 23, and 40), Hinks teaches that the Translation Table contains all the information needed to build editors that can simulate the target UI (User Interface) without having access to the sources or actual binary of the product to be translated (Col. 7, lines 22-26). In addition, Hinks teaches that the Translation Table encapsulates all the information that is known or can be derived from the various resources and stores them in a format which may be utilized by various editors (Col. 3, lines 16-19; compare with Claim 6 (and similarly Claims 23, and 40), ***"... the contextual information includes graphical user interface (GUI) rendering information"***).

In regard to independent Claim 14 (and similarly independent Claims 31, and 48), Hinks teaches that the Translation Table contains all the information needed to build editors that can simulate the target UI (User Interface) without having access to the sources or actual binary of the product to be translated (Col. 7, lines 22-26). In addition, Hinks teaches that the Translation Table encapsulates all the information that is known or can be derived from the various resources and stores them in a format which may be utilized by various editors (Col. 3, lines 16-19; compare with Claim 14 (and similarly Claims 31, and 48), ***"... receiving a sequential record of system events that occurred during an execution session of the computer program; executing an executable of the computer program, wherein the executable contains an introspective editor; and reproducing the system events from the sequential record to control execution of the executable"***).

In regard to independent Claim 35, Hinks teaches system (100) of Fig. 1, which includes a central processor (101), a main memory (102) (e.g., random-access memory or RAM), an input/output controller (103), a keyboard (104), a pointing device (105) (e.g., mouse, track ball, pen device, or the like), a display device (106), and a non-volatile or mass storage (107) (e.g., hard or fixed disk, optical disk, magneto-optical disk, or flash memory). Processor (101) includes or is coupled to a cache memory (109) for storing frequently accessed information; memory (109) may be an on-chip cache or external cache (as shown). System (100) may also be provided with additional input/output devices, such as a printing device (108), as desired. The various components of the system (100) communicate through a system bus (110) or similar architecture, as shown (Col. 5, lines 42-56; compare with Claim 35, ***"... a bus system; a memory connected to the bus system, wherein the memory includes a set of instructions; and a processing unit, wherein the processing unit includes at least one processor, wherein the processing unit executes the set of instructions to perform the acts of"***). Hinks also teaches that the Translation Table contains all the information needed to build editors that can simulate the target UI (User Interface) without having access to the sources or actual binary of the product to be translated

(Col. 7, lines 22-26). In addition, Hinks teaches that the Translation Table encapsulates all the information that is known or can be derived from the various resources and stores them in a format which may be utilized by various editors (Col. 3, lines 16-19; compare with Claim 35, ***"... receiving contextual information describing a visual context for the program's generation of the text; receiving the text from a first user interface text storage format; combining the contextual information with the text to form a visual representation of a display in the user interface; and displaying the visual representation"***).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 7-13, 15-17, 24-29, 30, 32-34, 41-47, and 49-51 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hinks in view of Liu et al. (hereinafter Liu, "Enhancing A GUI Event Recorder To Support The Creation Of User Documentation", Copyright 2000).

In regard to dependent Claim 7 (and similarly dependent Claims 13, 24, 30, 41, and 47), Hinks does not specifically teach that *the computer program is written in Java*. However, Liu teaches a Java-based GUI event recorder called Listen (p. 4, 2nd paragraph). It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Hinks and Liu providing the benefit of utilizing existing Java tools to record GUI events.

In regard to independent Claim 8 (and similarly independent Claims 25, and 42), Hinks fails to specifically teach *detecting when the executing computer program generates a user interface feature; and recording a description of the user interface feature*. However, Liu teaches Listen, a GUI event recorder which records information about the GUI event (what the user or system has done) and which GUI objects are being operated on in the interface (pp. 3-4, Sec. 3.1). It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Hinks and Liu providing the benefit of recording GUI events.

In regard to dependent Claim 9 (and similarly dependent Claims 26, and 43), Hinks teaches that the Translation Table encapsulates all the information that is known or can be derived from the various resources and stores them in a format which may be utilized by various editors (Col. 3, lines 16-19; compare with Claim 9 (and similarly Claims 26, and 43), ***"... associating the description with an item of text in a user interface text storage format"***).

In regard to dependent Claims 10-11 (and similarly dependent Claims 27-28, and 44-45), Hinks teaches that the Translation Table contains all the information needed to build editors that can simulate the target UI (User Interface) without having access to the sources or actual binary of the product to be translated (Col. 7, lines 22-26; compare with Claims 10-11 (and similarly Claims 27-28, and 44-45), ***"... the description***

includes information regarding the rendering of the user interface feature" and ***"... the user interface is a graphical user interface (GUI)"***).

In regard to dependent Claim 12 (and similarly dependent Claims 29, and 46), Hinks teaches that a (Export/Import) Resource Parser (330) parses the resource file (325) (such as a Windows .rc file) into a Translation Table (340), which is typically stored as a database table. The Translation Table (340) encapsulates all the information that is known or can be derived from the various resources and stores them in a format which may be utilized by various editors (Col. 7, lines 64-67; Col. 8, lines 1-5; compare with Claim 12 (and similarly Claims 29, and 46), ***"... the first user interface text storage format is a resource bundle"***).

In regard to dependent Claim 15 (and similarly dependent Claims 32, and 49), Hinks does not teach *responsive to a user input, execution of the reproducing suspending step* or *responsive to a second user input, resuming the execution of the reproducing step.* However, Liu teaches TAMOT, a task-modeling editor which takes as input a GUI event and eventually produces task models from which can be extracted online help (pp. 4-6, Secs. 3.1-3.2). Liu does not specifically teach a start and stop feature. However, it would have been obvious to one of ordinary skill in the art at the time of invention to have such a feature because it would have made the editing process easier. The benefit would have been to enable a pause feature in the editing of a GUI.

In regard to dependent Claim 17, Hinks fails to specifically teach *the system events include at least one of a keystroke, a mouse click, a mouse double-click, and a* *mouse drag*. However, Liu teaches Listen, a GUI event recorder which records information about the GUI event (what the user or system has done) and which GUI objects are being operated on in the interface (pp. 3-4, Sec. 3.1). Though Liu does not list specific events as claimed, it would have been obvious to one of ordinary skill in the art at the time of invention to assume that fundamentally, one or more of the claimed GUI events would have been recorded since it was commonly known that a Graphical User Interface, even the simplest one, would have features relying on a mouse click or similar action providing the benefit of GUI operation.

In regard to independent Claim 42, Hinks teaches system (100) of Fig. 1, which includes a central processor (101), a main memory (102) (e.g., random-access memory or RAM), an input/output controller (103), a keyboard (104), a pointing device (105) (e.g., mouse, track ball, pen device, or the like), a display device (106), and a non-volatile or mass storage (107) (e.g., hard or fixed disk, optical disk, magneto-optical disk, or flash memory). Processor (101) includes or is coupled to a cache memory (109) for storing frequently accessed information; memory (109) may be an on-chip cache or external cache (as shown). System (100) may also be provided with additional input/output devices, such as a printing device (108), as desired. The various components of the system (100) communicate through a system bus (110) or similar architecture, as shown (Col. 5, lines 42-56; compare with Claim 42, ***"... a bus system; a memory connected to the bus system, wherein the memory includes a set of instructions; and a processing unit, wherein the processing unit includes at least one processor"***). Hinks fails to teach *detecting when the executing program generates* *a user interface feature; and recording a description of the user interface feature.* However, Liu teaches Listen, a GUI event recorder which records information about the GUI event (what the user or system has done) and which GUI objects are being operated on in the interface (pp. 3-4, Sec. 3.1). It would have been obvious to one of ordinary skill in the art at the time of invention to combine the teachings of Hinks and Liu providing the benefit of sensing and storing GUI events for generation of online help.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to James H Blackwell whose telephone number is 703-305-0940. The examiner can normally be reached on Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Joseph H Feild can be reached on 703-305-9792. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

James H. Blackwell
06/22/04

Feild
JOSEPH FEILD
SUPERVISORY PATENT EXAMINER